PREDICTING THE TIME INTERVAL BETWEEN SHIFT PLACEMENTS AND SHIFT FULFILMENTS USING DIFFERENT REGRESSION MODELS

Student details

Name: Melchior Klop
Student number: u520396

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee

Supervisor: dr. B. Čule
Second reader: dr. D. Shterionov
External supervisors: David Telenga and Jesse Kirschner

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
14 January 2022
Words: 8710

Abstract

For online platforms that seek flex workers (employees that can determine their own work hours and work) to work certain shifts, there is always an uncertainty about the amount of time it will take to fill a particular shift with an flex worker. In this thesis, several machine learning algorithms will be used to predict the amount of time it takes to fill shifts with flex workers. Machine learning algorithms investigated in this thesis are a support vector machine, elastic net, artificial neural network, random forest and a linear regression model. By using job-related features derived from the database of Level.works (an online platform) and by using machine learning algorithms, the amount of time needed to fill a shift is predicted. Moreover, by means of feature importance and permutation importance, the most important features from the dataset were found, which have a large effect on the predictions of the models.

Data source and ethics statement

Work on this thesis did not involve collecting data from human participants and while writing the thesis the GDPR was respected. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data or code.
The code used in this thesis is not publicly available.

1 Introduction

The labour market is subject to major structural changes in the technological landscape as well as changes in the economic landscape. New forms of labour have emerged due to increased economic demand for flexibility and access to the internet and mobile phones. A recent and increasingly popular form of labour in the Netherlands that has crystallised out of technological and economic changes is labour through an online job platform (Wiel, van der Werff and van Kesteren, 2021). This relatively new form of work comes with opportunities for entrepreneurs, but also with many challenges. A problem and uncertainty for companies involved in directing the labour supply and demand through an online platform, is the time window between the placement of shifts (e.g. a single 8 hour shift in a warehouse) and its fulfilment. Companies in this sector will never know exactly why and how fast a particular shift is filled. A company that also encounters this problem on a daily basis is Level.works. Level.works is an online platform where companies can place shifts and flex workers can fulfil these shifts. An understanding of the time it takes to complete a shift and what features are important for a shift to be fulfilled can bring much more clarity to Level.works. The overall aim of this thesis is to predict the amount of time between the placement of a shift and the finding of a person (fulfilment of the shift) and to select the model with the least amount of error for this task. This will be done by applying different machine learning algorithms for the time prediction task and to compare the results with the results of a selected baseline model.

1.1 Practical and scientific relevance

Solving this problem has many practical implications for various parties. Level.works will probably benefit from a better understanding of their data, which features might be very important for accepting a work assignment and which might not. The second practical implication lies in the specialisation of the company. Level.works primarily cooperates with companies in the security, transport and logistics sector. Time is often of the essence in these sectors since something must be transported at short notice or someone must be found for an important security task. A better understanding of the amount of time needed to complete a work assignment can give the customers of Level.works more space and insight into their decision making process.

The described problem and its solution also has a scientific relevance. Much of the scientific literature that deals with online labour markets tries to map the size and influence of this market (e.g. Kässi & Lehdonvirta, 2018; Abraham et al., 2017; Kuek et al., 2015). Other studies that investigate online job markets usually try to scientifically examine the long-term advantages and disadvantages of this market (e.g. O'Farrell & Montagnier, 2020; Bellesia, Mattarelli, Bertolotti, & Sobrero, 2019). In many of these studies, the figures and study relate to online platforms in the United States or the United Kingdom, but not to online platforms in the Netherlands. There are many studies related to predicting time intervals (Curtis et al., 2018; Kuo et al., 2020; Martinez et al., 2021; Samba et al., 2018; Lin et al., 2019). However, there are not many related studies to time prediction in combination with data of an online platform. Moreover, most of the aforementioned studies relate to medical data and medical problems. A thesis with data of an online platform and a time prediction problem could therefore contribute significantly to research in this area.

1.2 Research questions

In order to solve the problem and contribute to scientific work in this area, the following research question is formulated:

How well do various machine learning algorithms predict the time interval between a shift and its fulfilment compared to a selected baseline model?

The emphasis in this question is therefore on applying several machine learning models and to compare the results of these models with a selected baseline model. The baseline model will be used first for predictions followed by more complex machine learning models. A logical, but also important question that naturally arises when answering the main research question is what features are important for predicting the amount of time between shift placement and shift fulfilment. An answer to this question could provide a better insight into our data and our machine learning models. The sub-question is formulated as follows:

Which features are important for predicting the amount of time between online shift placements and the fulfilment of the shift?

The answer to this question is sought through the use of feature importance and permutation importance, which will be discussed in the methods section of this thesis.

2 Related work

The related scientific work linked to the problem presented in this thesis can be divided into two main scientific areas. On the one hand, there are scientific studies related to online job platforms and, on the other hand, there are studies related to making predictions about time. The latter area will be the most relevant for this thesis, as its methods, evaluation, problems and general course of scientific research are most similar to the research and problem in this thesis. Related research on predicting time intervals tries to predict a time interval as well as possible by means of a dataset and multiple predictive models. Thus, more attention will be given to time prediction literature in this thesis. The scientific research on online job platforms will be dealt with first in this section of this thesis. This is done to reflect the broader context and important features of the problem.

2.1 Related work on online platforms

A striking phenomenon in academic research on online job platforms is that there is no umbrella term to describe an online platform where services or jobs are offered. This is probably due to the fact that many different services are offered online and therefore the line is blurred for a clear term. Pontgratz (2018) shows in his work that there are different ideas about what to call online job platforms and that there is no scientific consensus on a fixed term. A number of important differences are shown by Codagnone et al. (2016) in their research for designating online job markets. The most crucial difference is the physical or online component of the service. If the service is performed entirely online and if there is an online platform that directs this traffic, then we speak of "Online labour markets" (Codagnone et al., 2016). If an online platform directs the supply and demand for physical services (e.g. food delivery), we speak of "Mobile labour markets" (Codagnone et al. 2016). Codagnone et al. argue that even in this distinction, there is much overlap between the two services. Therefore, this thesis will not make use of this distinction, but will refer to the term online job platform. This term covers all online platforms that offer jobs or services (both physical and online).

There are a number of issues that academic literature is addressing regarding online job platforms. The size and growth of this market, its advantages and disadvantages and predictive models for aspects of online job platforms. The size and growth of online job platforms is a very active field of research (e.g. Kässi & Lehdonvirta, 2018; Abraham et al., 2017; Kuek et al., 2015; O'Farrell & Montagnier, 2020). For example, Abrabam et al. (2018) tried to map the size of the online job market on the basis of historical household data (e.g. U.S. Census Bureau survey data) and administrative data. One conclusion they drew was that it is very difficult to capture the size of this market based on household data and other administrative data sources. A consequence of this is that the actual growth of labour is not well measured and the actual number of people working is actually much higher than assumed (Abrabam et al., 2018). This issue is confirmed in the study of O'Farrell & Montagnier (2020). The authors stress the difficulty of determining the actual number of flex workers and their characteristics with current statistical sources (O'Farrell & Montagnier, 2020). The work of Kässi & Lehdonvirta (2018) tries to address this problem by moving away from traditional statistical sources for measuring the labour market. The authors propose a new indicator that can be used to visualise the size and growth of online job platforms. This new indicator, which is called the "Online Labour Index" in the article, measures the actual number of jobs and other relevant data to capture the size of this market (Kässi & Lehdonvirta, 2018). An important note here is the fact that this only concerns online flex workers and not flex workers that also perform physical services.

The final section related to relevant scientific literature on online job platforms is a selection of works that provide insight into the important components of an online platform and predictions regarding features important to platform holders. It is notable that in the predictive literature focused on online platforms, no works are present that deal with time interval prediction. Themes that do emerge frequently in this specific scientific field include predictions about the number of visitors to the platforms, the likelihood of being hired for a job and analyses of which elements of a platform are important in the selection process of an employer or customer. An observation when reading these works is that, once again, they are mostly focused on online job platforms that offer online services. This is also noted in the work of Codagnone et al. (2016), they describe online platforms with physical services as an area of scientific research that is still open for investigation (Codagnone et al., 2016).

A scholarly work that provides insight into data from online platforms from the employer's point of view is the work of Kokkodis et al. (2015). Through different probabilistic models (e.g. logit model) and an analysis of the features they have chosen in their dataset (from both employee and employer), they want to calculate the hiring probability of platform workers. Based on more than 600,000 job applications from oDesk.com they have obtained results in terms of feature importance and predictive power of their models (Kokkodis et al., 2015). Kokkodis et al. extract from the coefficients of their models the features that have a high effect on the hiring probability and use the area under curve evaluation metric to compare the created probabilistic models with their baseline model. The features that appear to have strong effects are country, profile completeness and previous collaborations. Notable features that had an unexpectedly weaker effect on hiring probability were invoice price and rating (Kokkodis et al., 2015).

2.2 Related work on predicting time intervals

The scientific field of time interval prediction is a very relevant and active one. An overarching theme in this field is the use of multiple machine learning models to make a comparison with a selected baseline model. Many studies in this direction have a predictive problem with a medical or business background (e.g. Curtis et al., 2018; Samba et al., 2018). As described earlier, the selected models in this thesis are mainly based on these studies. The study by Kuo et al. (2020) is a great example and reference point for today's popular regression models and the research methods commonly used in this research field. The authors attempt to calculate future waiting times of patients in an emergency department of a Hong Kong hospital using four regression models. They use two sets of features to implement the models, which can be divided into main features and features that provide additional information (Kuo et al., 2020). As a baseline model, the authors use a linear regression model, which is also frequently used in other studies in this research field (e.g. Curtis et al., 2018). The other three models used in this study are a support vector machine, a gradient boosting machine and an artificial neural network. The hyperparameters of the latter models were tuned via grid search (Kuo et al., 2020). The amount of hyperparameter units on which the grid search is set up does appear to be unclear in this study. In the data cleaning phase of this study there were a number of important observations. An important observation that the authors made when cleaning the data was the observation of outliers (abnormal values) and removing them from the training set. In other studies all outliers were removed (e.g. Martinez et al., 2021). Due to the nature of an emergency department, times vary greatly between patients and therefore the authors emphasise that removing the outliers was necessary for their predictions (Kuo et al., 2020).

The results of the study by Kuo et al. (2020) were also very interesting. After data cleaning and engineering of a number of features, the authors concluded that the baseline model with the implementation of both features was outperformed by the other three models. Based on the coefficient of determination (R^2), mean squared error (MSE) and root mean squared error (RMSE), the results of the models were compared with the results of the baseline model. They also noted that it was difficult for the models to identify important features. This was reflected in the high bias of the predicted times (Kuo et al., 2020). Another important observation from this study, and one that the authors certainly stress, is the fact that a machine learning model for time prediction needs large amounts of training data and also requires a lot of domain knowledge. The domain knowledge is needed to identify or create the right features for better model predictions (Kuo et al., 2020).

The idea that domain knowledge is needed for better predictions is confirmed by the study of Curtis et al. (2018). The aforementioned study includes a similar investigation of waiting times in a radiology facility in Massachusetts. The authors of this study substantiate the selected features for their machine learning models based on their domain knowledge of radiology facilities. Based on as many as 40 different predictor variables, the authors compared nine different machine learning models and a linear regression model (Curtis et al., 2018). It is noteworthy that the study does not describe how the hyperparameters were tuned, but does extensively describe the feature selection. Another interesting fact is that the authors of this study had no trouble identifying the important features, in contrast to the study by Kuo et al. (2020). Among the nine machine learning models were a neural network, support vector machine, regression tree and elastic net. The machine learning models were compared for the best performance based on their RMSE and R^2. It was notable, according to the authors, that the elastic net model performed best among all the models. The authors argue that an elastic net is based on a simple linear regression and is therefore computationally quite simple. It is therefore remarkable that an elastic net performs better than other much more complex models with many hyperparameters (Curtis et al., 2018).

A study that reached a similar conclusion about the complexity of a machine learning algorithm and the result is the study by Martinez et al. (2021). Martinez et al. conducted research on predicting operation times in a hospital. Based on a large dataset of medical data over a period of 14 years, they trained four different machine learning models to predict time as accurately as possible. The dataset they used contained a lot of categorical data (e.g. type of anaesthetic and day of operation) that is not implementable in, for example, a support vector machine. They solved this by changing the categorical data to numerical values using one hot encoding and sequential encoding. In addition, they removed all abnormal operation time records from the database that could influence the results. The most important features in their dataset were chosen based on an analysis of variance, due to the size of the database. For the selected machine learning models, they selected a regression tree, support vector machine, bagging regression tree and a linear regression model. The hyperparameters were manually tuned for each algorithm with 10 fold cross-validation (Martinez et al., 2021). As described earlier, there was an aspect of Curtis et al. (2018) study that corresponded. For both studies, a relatively computationally simple machine learning model performed best compared to other models. Based on the RMSE of all models, the bagged regression tree was found to perform best. Based on the aforementioned results, it seems that tree-based algorithms are a good basis for time-related predictions (Martinez et al., 2021).

A work that offers a different insight and does not deal with medical data, but with a more business-like problem is the work of Samba et al. (2018). This work is very interesting because it uses multiple data sources to predict the download time of a file (e.g. movie clip) in a mobile network. Samba et al. use different machine learning models and features related to mobile networks, such as the number of users at a given time, but also features related to other sources, such as the strength of a radio signal. Moreover, they distinguished themselves from other studies related to download time over a mobile network because they try to do so without historical metrics (Samba et al., 2018).

The dataset the authors use to predict download time is large (around 20,000 entries) and is mainly collected from a mobile network in France. Samba et al. (2018) use a different method to deal with missing data in their dataset. Unlike the work of Kuo et al. (2020) who used an imputation method to fill missing records or the work of Martinez et al. (2021) where the missing records were simply deleted, Samba et al. use a method that spreads the missing data over several datasets. The convenience of this is that the features and their values remain correct, as the datasets are specially designed for each missing value (Samba et al., 2018). Samba et al. use this method to avoid the loss of useful data. In the study by Martinez et al. the removal of records with missing information led to a lot of data loss. Finally, an interesting fact was which model could best predict the download time. The model with the best R^2 and the lowest median absolute error ratio after adding all predictors was the random forest model in their research (Samba et al., 2018).

The last article that is related to time prediction and that uses different machine learning models to predict patient waiting time as best as possible is the article by Lin et al. (2020). Using a dataset from an ophthalmology clinic, they predicted the waiting times for different departments for the clinic as well as possible. After data pre-processing, cleaning and training the models, the best model based on RMSE and R^2 was found to be the random forest as in the study by Samba et al. (2018). The two worst performing models were the support vector machine and gradient-boosting machines. This is remarkable since the two aforementioned models were usually in top or middle of best predicting models in other related works (e.g. In the study of Curtis et al. (2018) the support vector machine had a low RMSE compared to the other models). The most important features of the model that had a strong effect on the random forest were the ophthalmologist and pupil dilation (Lin et al., 2020).

3 Methods

3.1 Chosen methodology

In this section, the general approach behind the thesis is explained. The methodology of the time prediction related literature is adopted (applying different machine learning models on a dataset to predict a continuous time related feature). In this thesis multiple machine learning models will be computed and compared in order to select the most accurate model among them for the specified task. The final models selected for this thesis are also based on the scientific literature on time-interval prediction and are a compilation based on all models in related work. The idea behind the selection was to choose a wide variety of models with different embedded properties. The machine learning models chosen for comparison are an elastic net (EN) (Zou & Hastie, 2005), support vector machine (SVM) (Hearst et al., 1998), feed-forward artificial neural network (ANN) (Bishop, 1995), random forest (RF) (Breiman, 2001) and as a baseline a linear regression model (LR) (Hocking, 1983). LR was chosen as the baseline model because of its simplicity, interpretability and good performance as baseline model in related work. For example, four of the five scientific works discussed in related work on predicting time intervals use a LR as baseline model with good results. As an example, in the work of Curtis et al. (2018) the LR is the second-best model out of 10 models. The major disadvantage of the LR is that the simplicity of the model does not capture certain correlations and that outliers can significantly influence the outcome (Samba et al., 2018).

The second mentioned model is the EN and is a regularisation algorithm that uses two popular penalties (from the lasso and ridge method) during the training phase to penalise the model. This model was chosen because the study by Curtis et al. (2018) had very good results with the model. Moreover, this model is popular among similar studies (e.g. Lin et al., 2020). The third model chosen is the SVM. This model was chosen mainly because the model is known to handle high dimensionality data well (*Sklearn.Svm.SVR*, 2021). Although the model is mainly used for classification tasks, the model can also be used well for regression problems. This can be seen in the aforementioned literature, where four out of five studies use a SVM as comparison model. Another model chosen to implement in this thesis is RF. This model was chosen because several related works had great predictions with the model. In both the study by Samba et al. (2018) and Lin et al. (2020) the most accurate model was the RF and in other previously mentioned works the model performed as one of the best models.

The last model implemented for this thesis is the ANN. This model is chosen as a complex counterpart to the two simpler models (linear regression and elastic net). Samba et al. (2018) describe in their work that an ANN is able to handle more complex correlations of features. However, a disadvantage is that the model is more prone to overfitting (Samba et al., 2018). Overfitting is a phenomenon whereby the model learns the training data too well, including irrelevant data points in the training data, making the model less generalisable and reducing its performance on the test data (Jason Brownlee, 2019).

For the sub-question on feature importance, a fitted RF (the same RF that is used for predictions in this thesis) was chosen to compute the feature importance. By means of the built-in function of scikit-learn feature importance it is possible to discover the most relevant features in the RF. The aforementioned method is also used in the study of Lin et al. (2020) to discover the most important features in their dataset. The feature importance of an RF is calculated based on the feature importance of all trees in an RF. For all trees in the RF, the most important nodes in each decision tree are chosen on the basis of variance reduction measured in mean-squared-error (MSE) defined as:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i-\mu)^2$$

where y_i is the label of an observation, μ the mean of an observation and finally N the number of observations (Ronaghan, 2018). The feature of a database that leads to the largest decrease in the impurity of a split in a decision tree (impurity of a split refers to how much MSE is measured when data is split on a certain feature/node in a decision tree) divided by the likelihood of reaching the node is considered the feature with the highest feature importance (Ronaghan, 2018). The importance is defined as:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k\in all\ nodes} ni_k}$$

where ni_j is the importance of a certain node and fi_i is the feature importance of a feature in a single decision tree. For feature importance on a random forest scale, the formula is defined as:

$$RFfi_i = \frac{\sum_{j\in all\ trees} normfi_{ij}}{T}$$

where T is the number of decision trees in the random forest, $RFfi_i$ is the feature importance of feature i computed across all decision trees and finally $normfi_{ij}$ is the feature importance of feature i transformed to a value between 0 and 1 in tree j (Ronaghan, 2018). In addition, a permutation based feature importance will be computed to compare its results with the results of the feature importance. This will also be done with the scikit-learn library. The permutation based feature importance is calculated on prediction error after a feature is randomly shuffled of a fitted model (RF in this thesis). Internal model relations are thus not calculated and the decrease in accuracy of the model determines which features are important (Aldrich & Auret, 2010). The formula for permutation importance is formally defined as:

$$i_j = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j}$$

where i_j is the permutation importance of a database column j, s the accuracy score (R^2 for regression) and k the number of repetitions (*Permutation Feature Importance*, 2021).

To examine the accuracy of our predictions, this thesis looked at two evaluation metrics. For each model, the coefficient of determination (R^2) and the root-mean-squared error (RMSE) were calculated based on the test set. The R^2 and RMSE are calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\hat{y} - y_i)^2}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

where $\hat{y}_i$ is the predicted time difference between shift placement and shift fulfilment, y_i is the actual time difference between shift placement and fulfilment, $\hat{y}$ is the mean of the target variable computed of the test set and N is the number of observations (Samba et al., 2018; Martinez et al., 2021). Logically, the best model is the one with the lowest RMSE and the highest R^2 on the test set. Four of the five studies related to time-interval prediction use RMSE and another evaluation metric to compare the models. All studies use two evaluation metrics (other interesting metrics included computational time and the median absolute error ratio) to determine the accuracy of their predictions. RMSE and R^2 were chosen for their simple interpretation. The explanation for using two evaluation metrics is given by Samba et al. (2018). Two evaluation metrics can counteract the bias of a single evaluation metric. Indeed, a model may perform very well on one evaluation metric but very poorly on another (Samba et al., 2018). A combination of evaluation metrics is therefore chosen in this thesis.

For the evaluation of the feature importance and permutation importance this thesis will look at the feature importance and permutation importance at a random forest level (both formulas described earlier in this section). Feature importance describes which feature will cause less variance in the model (mean impurity decrease) and permutation importance describes which feature increases the performance of the model (Lewinson, 2019).

4 Experimental Setup

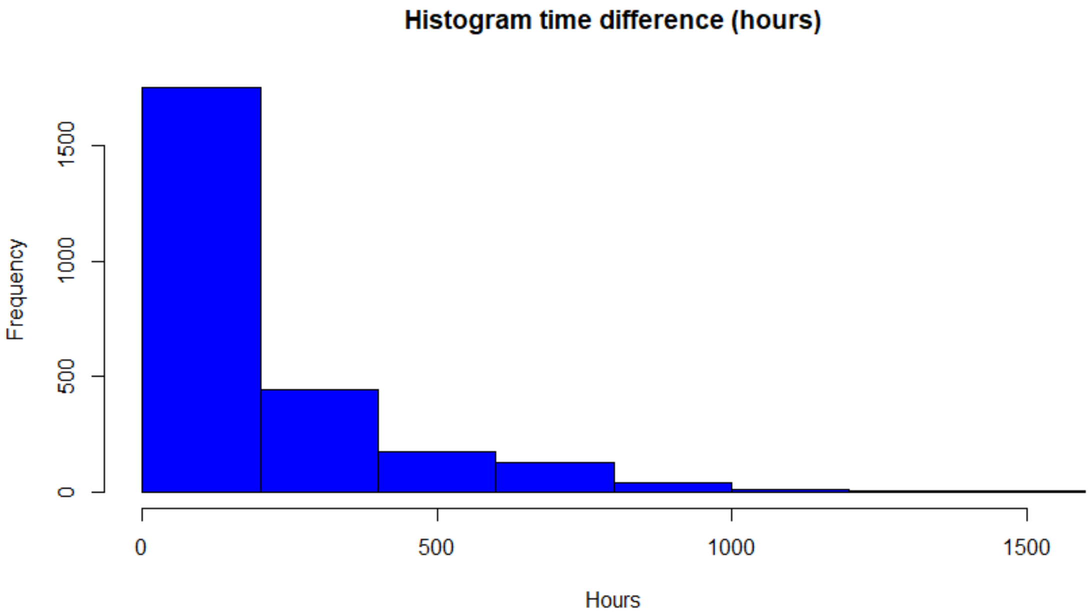
4.1 Data pre-processing and cleaning

The dataset used for this thesis is owned and provided by the company Level.works. Therefore, the dataset is not publicly accessible. The dataset is an extraction of a part of the Level.works SQL server and contains data of the months July, August and September 2021. The variables in the dataset are job and time related and will be described in detail later in this section. The dataset was provided in three parts and to join the three parts, the R programming language with the dplyr package was used (*Dplyr Package - RDocumentation*, 2022). The raw dataset consists of 5454 observations and 33 variables. With the available variables, the target variable was created. The target variable, which is the time difference between shift placement and shift fulfilment, had to be created. First, the data had to be filtered on all shifts that were accepted, shifts that were completely filled and finally the latest time that completely filled the shift. The time difference was then expressed in hours. The next step in the pre-processing step was to remove all columns that were not relevant (e.g. columns containing data for the administrator of the SQL server) or contained not relevant company information. Variables were also removed that were no longer relevant due to the aforementioned filter actions, such as acceptance state or the id of a particular shift. The final selection contained 9 variables and the number of observations was 2551. The nine variables used for training the model were shift_createdAt, shift_start, shift_end, max_claims, hourly_wage, fine_noShowJob, job_function, job_location and the target variable diff_time. A clear description of the nine variables can be found in table 1 in the appendix. The data then had to be adjusted further. All variables related to time were in POSIXct format and had to be transformed to a machine readable format with the lubridate R package (Grolemund & Wickham, 2011). Shift_createdAT, shift_start and shift_end were thus transformed to integers, which represented the hour in the POSIXct format. Max_claims, hourly wage and fine_noShowJob were already in numeric format and did not need to be modified. Two categorical variables job_location and job_function posed a bigger problem during the data pre-processing step. Both variables contained string data with a lot of noise (e.g. additional information of certain locations) that had to be cleaned up with R's tidyverse collection. After cleaning both categorical variables there were approximately 28 different locations and 35 different job functions left.

For the exploratory data analysis, we first looked at the data distribution of the target variable. Figure 1 shows that the data of this variable was positively very skewed.

Figure 1

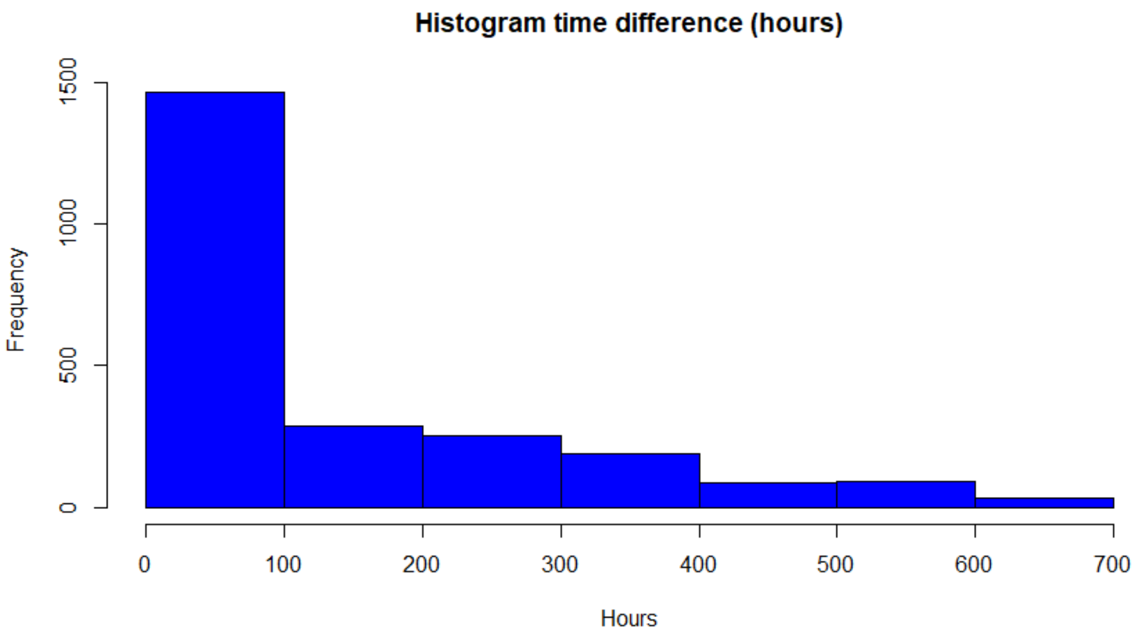
Histogram of target variable distribution (outliers not removed)



This can be explained by the large variability in time of shift fulfilments. Some shifts are filled within minutes, but some shifts take multiple days to get filled. This is also a problem in the relevant literature on time prediction. In the research of both Lin et al. (2020) and Martinez et al. (2021) it can also be seen that sometimes the waiting time in an emergency department or clinic is very short and other times it is exceptionally long. One method they used, was to omit the abnormal values from the data and thus the predictive model. This was done in the aforementioned studies to ensure the statistical interpretation and not to influence the predictive models too much with outliers (Martinez et al., 2021). In order to guarantee the statistical interpretation in this thesis as well and not to train the model on outliers, the deviating values of the target variable were first identified with a boxplot. A logical next step was then to remove the relevant outliers from the dataset. Lin et al. used a certain time threshold to keep times within an hour and everything beyond that time was excluded (Lin et al., 2020). In this thesis, the high variability of times was kept (after removal of the outliers) in the target variable distribution as shown in figure 2, since these times should also be predictable. The threshold for the target variable distribution is therefore 700 hours.

Figure 2

Histogram of target variable distribution (outliers removed)



Another problem that can lead to less good predictions is the aforementioned skewness of the data distribution of the target variable. An option to change the skewness is a log transformation. A log transformation on data that is skewed heavily to the right can give a more normal data distribution and this can lead to better predictions and statistical interpretation (West, 2021). However, a log transformation is subject to much debate in academia. In a less recent article by Feng et al. (2014) on the implications of log transformed data, the log transformation is strongly discouraged for dealing with right skewed data. According to Feng et al., a log transformation of a feature may even increase skewness in cases where the data does not contain a log normal distribution. Moreover, log-transformed variables are difficult to use, as the interpretation is not the same after the log transformation (Feng et al. 2014). However, other works also recommend using it, especially if all numbers are positive and the data does not contain too many zero values (e.g. Ekwaru and Veugelers, 2018; West 2021). In this thesis, a log transformation was considered, but ultimately not chosen due to the arguments listed in the work of Feng et al. The next step in the pre-processing process was to find and treat all missing variables. A notable conclusion was that the dataset had no missing values, so no imputation method or removal was needed.

The last, but very important phase of the pre-processing step was to change the categorical variables (job_location and job_function) to machine learning readable format. A number of options were considered in this thesis for the transformation of the categorical variables. Based on the scientific literature, one-hot encoding seemed to be the best option. The research by Martinez et al. (2021) and the research by Lin et al. (2020) both use a method of one-hot encoding to transform the categorical data. However, to avoid the dimensionality problems (too many input features), a different encoding was chosen in this thesis. The two categorical variables both contain around 30 different categorical labels. With the one-hot encoding method, this would therefore result in 60 new columns in the data set. The research by Martinez et al. also used another encoding scheme, namely sequential encoding. This option was not chosen because correlation biases could be formed between the encoded variables, which the study by Martinez et al. also pointed out as a problem (Martinez et al., 2021). In this thesis, a form of target encoding was used, namely CatBoost encoding. The work of Prokhorenkova et al. (2018) shows that this form of target encoding is well suited to features with many categorical labels. Catboost target encoding, like other target encoders, uses the mean of the target feature, but also uses the structure of the data. It is a supervised target encoder that prevents target leakage, which is often a problem with other target encoders (Grover, 2019). Target leakage is the phenomenon where information from the target feature is leaked to the training data, causing the model to quickly overfit (Kuhn and Johnsen, 2019). To implement the encoder, the data was split beforehand with the scikit-learn train-test-split function into a training set and a test set. For the size of the training and test set a ratio of 80 to 20 was used with 80 percent of the data in the training set and 20 percent of the data in the test set (*Sklearn.Model_selection.Train_test_split*, 2021). The Catboost encoder was trained on the train data and then used to change the categorical features in the train data. To prevent target leakage, the same trained encoding based on the training set was used to transform the categorical features of the test data set.

4.2 Experimental procedure

All models chosen in this thesis have been implemented and tuned with the python programming language. The python programming language has been used on a local server. For the implementation of the models, as described earlier, 80 percent of the dataset was randomly assigned to a training set and 20 percent to a test set. All models were trained on the training set and tested for evaluation on the test set, where the RMSE and R^2 were calculated for each model. The first implemented model was LR. This model served as a baseline model and was not tuned to keep the model simple.

The second implemented model is EN. EN is implemented with the scikit-learn library, just like the LR. The EN is tuned by means of grid search with a five-fold cross-validation. In this thesis, all models tuned with grid search were also tuned by five-fold cross-validation. The hyperparameters that were placed in the grid search were alpha, l1_ratio and max_iter. Alpha is a float number that multiplies the penalty measures of the model. This hyperparameter was set from 0.001 in magnifying steps to 1 in grid search. The l1_ratio is a penalty measure for the EN and this hyperparameter was placed in grid search in steps of 0.1 from 0 to 1. The last hyperparameter max_iter is simply the maximum number of iterations (*Sklearn.Linear_model.ElasticNet*, 2021).

The third model implemented is the SVM, also called support vector regression in the official documentation (*Sklearn.Svm.SVR*, 2021). The SVM is a popular algorithm among the studied works and again, this machine learning model is also tuned with grid search. An important observation in the related work was that all studies using it set the hyperparameter kernel to linear (e.g. Martinez et al., 2021; Lin et al., 2020). In this thesis, for the above reason, it was decided to keep the kernel on linear and add two hyperparameters in grid search. The penalty measure C and the loss function epsilon were placed in grid search in values from small to large and the other hyperparameters were set at default.

The fourth implemented model is the RF. The only hyperparameter tuned is n_estimators, i.e. the amount of trees in the algorithm. This value is kept at 100, since above 100 trees the computational time increases, but with minimal performance improvement (Oshiro et al., 2012). The RF is also used to answer the sub-question about feature importance and permutation importance. The study of Lin et al. (2020) also use an RF for feature importance and that was the main reason for using RF in this thesis as fitted model for the sub-question. The RF is also chosen as fitted model for permutation importance (since it can be computed for any fitted model) to make a relevant comparison with the results of the feature importance that was computed with the RF.

The last implemented model and also the most complex model is the feed-forward ANN. This model was implemented using keras python library. ANN consists of several elements. The first part for creating an ANN is constructing the sequential model. A component for constructing the model is the amount of hidden layers that are present in the model. The number of hidden layers in an ANN is still subject of much discussion. For example, in the related work of Samba et al. (2018), a single hidden layer is used for predictions of their regression problem. Since there is no consensus on the amount of hidden layers, in this thesis we chose to run the model with a single hidden layer as well as with two hidden layers. The number of input variables for the first layer is 8 and equal to the number of input features in our dataset and in the output layer the number of neurons (units) is one, since we want to calculate one continuous value. The number of neurons (units) in the other layers is determined by means of grid search. In the hidden layers, the activation function reLU was chosen manually. The other activation functions are more suitable for classification tasks and the tanh function is computationally heavier than the reLU activation (Tharsanee et al., 2021). For the error function in the network we chose mean squared error, since we are working with a regression problem and we want to minimize this metric (Reed, 1999). The adam optimiser function has been chosen as the optimiser that adjusts the learning rate and weights in a neural network (Doshi, 2019). The aforementioned optimiser has also been used in the work of Kuo et al. (2020). The final hyperparameters, such as batch_size (number of samples in network) and nb_epoch (number of iterations over dataset) were tuned with grid search.

Table 2

The final selection of hyperparameters tuned with grid search

ML algorithm		Hyperparameters		
LR		None		
SVM	C: 0.1	Epsilon: .1	Kernel: linear	
EN	Alpha: .01	L1_ratio: .1	Max_iter: 50	
ANN	Batch_size: 10	Nb_epoch: 50	Optimizer: adam	Units: 40
RF	N_estimators: 100			

Note. LR: Linear Regression, SVM: Support vector machine, EN: Elastic Net, ANN: Artificial Neural Network and RF: Random Forest.

5 Results

In this first section of the results, the results of the applied machine learning models will be discussed. The RMSE and R^2 of all models are shown in table 3. The algorithm that performed best based on R^2 and RMSE was the RF. The target variable had an R^2 value of .623 and could account for around 60 percent of the variance of the target variable in the model (time interval between shift fulfilment and shift placement). The RMSE of the RF model was 98.70 hours and overall prediction error of all models was around 112 hours. The baseline LR performed exceptionally well with a RMSE of 112.85 and a R^2 of .508. This was not expected, since it was assumed in related work that more complex machine learning models can detect more complex relationships between variables in comparison to a LR model (Kuo et al., 2020). In addition, the LR model was in all related work outperformed by several machine learning algorithms by a substantial margin. In this thesis only the RF outperformed the baseline model by a substantial margin and the EN outperformed the baseline with a very small margin of RMSE less and slightly higher R^2. The relative large prediction error of all models is likely due to the fact the data was positive skewed and contained extremely low values (fractions of hours) and large time values (up to 700 hours) even after removing the outliers.

The R^2 is somewhat similar to prior research. For example, Lin et al. (2020) could explain 37 percent of the variance, Curtis et al. (2018) with their best model around 59 percent and Kuo et al. (2020) around 60 percent. Samba et al. (2018) could explain around 81 percent of the variance. The RMSE did not correspond to prior research. This is because each related scientific work used a different time scale and threshold for outliers (e.g. Lin et al. used a threshold of an hour and that was relevant for waiting times prediction). The performance of the RF was within our expectations as the RF also performed best in the study of Lin et al. and Samba et al. In other time prediction related studies where the RF was used, it always performed in the top best predictive models as well (e.g. in the study of Curtis et al. the RF was the third best predicting model). The two models that predicted less accurately were the ANN and SVM. The result of the SVM was somewhat expected. The model had the highest RMSE and lowest R^2 of all models, as shown in Table 3. In related work the SVM performed alternately. In the study by Lin et al. the SVM was also the least accurate model and in, for example, the work by Kuo et al. it was one of mid-range models. The result of the ANN was within expectations, since this model also performed poorly in other related studies. An explanation for this could be that the model quickly overfits due to the two hidden layers in the model. For this reason, the model was ran once with one layer and once with two layers (as earlier mentioned in section 4.2 experimental setup). The method with a single hidden layer yielded an slightly lower RMSE and an slightly higher R^2 seen in table 4.

Table 3

RMSE and R^2 of all machine learning algorithms

ML algorithm	RMSE	R^2
Random Forest	98.70	.623
Elastic Net	112.73	.508
Linear Regression	112.85	.507
Artificial Neural Net	118.87	.453
Support Vector Machine	119.02	.451

Note. Best ML algorithm is on top.

Table 4

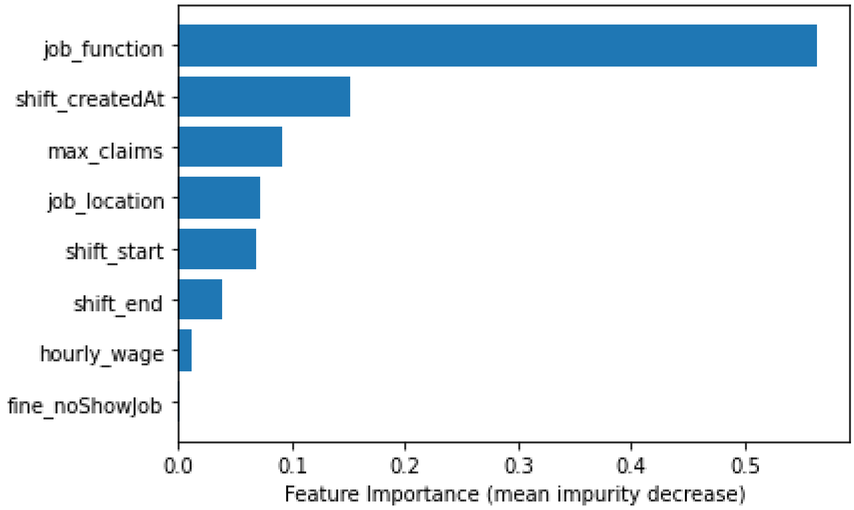
RMSE and R^2 of Artificial Neural Net with one hidden layer

ML algorithm	RMSE	R^2
Artificial Neural Net	118.41	.457

In this section of the results, the results of the feature importance of the RF algorithm will be presented and discussed further. The results of the RF feature importance and permutation based feature importance are shown in figure 3 and 4, respectively. The first thing to notice in the figure is the high feature importance of the feature job_function. One explanation could be that the feature importance is biased by the amount of unique values in the job_function feature (the categorical feature had 35 unique labels). After all, the official scikit-learn documentation also warns for misleading results concerning features with many unique values (Feature Importances with a Forest of Trees, 2021). The official scikit-learn documentation of feature importance therefore recommends comparison with a permutation-based feature importance to examine the bias of features. Looking at figure 4 we see that the permutation based feature importance gives an even higher feature importance value to job_function. As previously described in methods section 3.1, internal relations of models are not included in permutation importance. Thus, job_function turns out to be a very important feature for the regression models in this thesis. The second feature with a high feature importance is shift_createdAt. This feature is also the second most important feature in the permutation feature importance. So the time of when a job is placed is important for the predictions of the model and has a strong relationship with the target feature. The feature job_location also has a high feature importance (which was to be expected), but also max_claims appears to have a high feature importance. An explanation for the high feature importance of max_claims could be that a shift that needs a single flex worker gets filled faster than a shift that needs a large amount of flex workers to get fulfilled. A flex worker may think that if he/she does not fill the shift quickly enough, someone else will do it. Two features that showed surprisingly little feature importance in both figures were hourly_wage and fine_noShowjob. Kokkodis et al. also noted a weaker effect of hourly wages for another phenomenon (hiring probability), where other effects such as country and profile had a much stronger effect on hiring probability (Kokkodis et al., 2015).

Figure 3

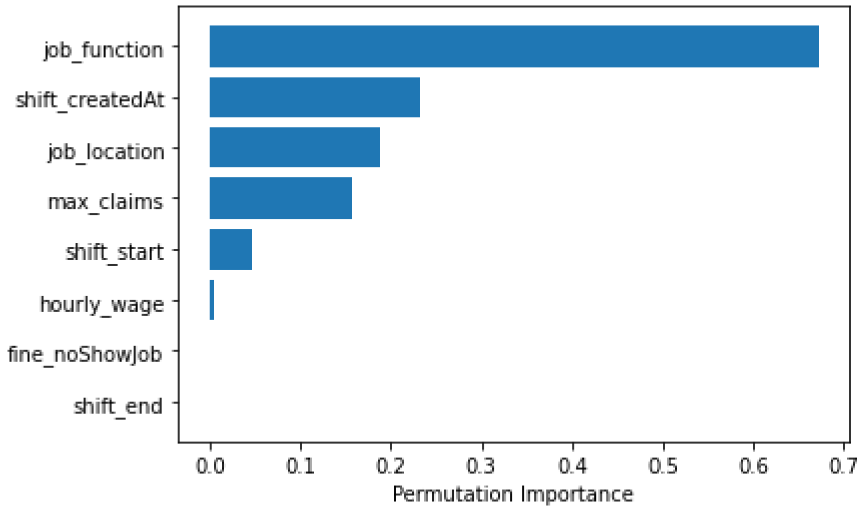
Random forest Feature importance



Note. Number of trees = 100. Best features are on top

Figure 4

Permutation based feature importance



Note. Permutation based on a trained random forest model. Best features are on top.

6 Discussion

The central goal of this thesis was to evaluate different machine learning algorithms and their predictions of the time interval between shift placement and fulfilment. A logical sub-question was to find the most important features in the dataset that had a relatively large influence on the predictions. The different applied machine learning algorithms had an average RMSE of 112.43 and an average R^2 of around .508. Among the different algorithms, the RF proved to be the most accurate at predicting the time interval (RMSE = 98.70 and R^2 = .623). Even with the large variability in the distribution of times (from a few minutes to 700 hours), the machine learning models were able to make relatively good predictions (around 5 days error rate). Although the predictions are not very specific (predictions can be off by several days), the predictions of the machine learning models (especially the RF) can provide valuable information and reduce uncertainty for the customers of Level.works. Customers can use the forecasts to make long-term shift placement plans (with error rate in mind) and thus reduce planning stress. Moreover, by using the forecasts they can save time and use this time for more priority issues.

The results of this thesis correspond to results of related work. As mentioned earlier in the results section, other related work had similar results with the models applied in this thesis. RF and EN were two models that had the lowest RMSE and the highest R^2 of all applied models. In related work, the EN and the RF and one other tree-based algorithm (bagged trees) also gave the best predictions. Martinez et al. (2021) emphasize in their discussion section of their work that previous time prediction works also had good results with algorithms that are based on linear methods or that are tree-based. The aforementioned algorithms contain properties that are very suitable for time predictions (Martinez et al., 2021). Samba et al. explain in their work that the properties of an RF algorithm are well suited for time prediction, as these models pick up more than just linear relationships between the target feature and other features in the model (Samba et al., 2018). However, this does not explain why the algorithms based on linear regression methods also give good results. The work of Curtis et al. (2018) on the contrary shows that the properties of an EN (regularisation, penalty method and linearly separation of datapoints) are well suited for predicting time, which can be confirmed by the good predictions in other related work (e.g. in the work of Lin et al. (2020), the EN had the second best predictions among the investigated models).

Models that made significantly less accurate predictions were the SVM and ANN. One explanation for the inferior performance of the ANN, might be that the model overfitted the data, which led to inferior predictions. This is also pointed out as a problem by Samba et al. (2018) who argued that mainly feed-forward neural nets are prone to overfitting (Samba et al., 2018). The predictions of the SVM can perhaps be explained by its properties. It may be that the properties of the SVM are not suitable (such as separating data points by means of a hyperplane) for the features and target variable present in this thesis.

For the related sub-question, feature importance and permutation importance were used to extract the most important features from the dataset. The four most important features for the RF model were job_function, shift_createdAt, job_location and max_claims. Job_location and job_function are very important features for both feature importance models. The influence of these features can perhaps be explained by the fact that certain locations and functions are more popular among flex workers for a shift. As described earlier in the results section, Kokkodis et al. (2015) derived that location and previous collaboration were important features in their models. The influence of shift_createdAt can perhaps be explained by the fact that some shifts are very popular and, once created, are filled in a short time.

In this section of the discussion, the limitations of this thesis will be explained. The main limitation of this thesis will probably be the amount of data. The data consisted of three months, of which a lot was removed by data transformations and outlier removal. A larger data set could provide better predictions for future studies in this research field. Related work also indicates that machine learning models often perform better with more data (Lin et al., 2020 and Martinez et al., 2021). A second limitation is the number of features. More domain knowledge is needed for feature engineering and feature selection. Kuo et al. (2020) emphasise in their work that domain expert knowledge significantly affects the predictive capabilities of machine learning models (Kuo et al., 2020). New features created with a domain expert that, for example, provide even more information about a particular job (a rating of the shift, for example) or a feature that contains keywords about a particular shift could perhaps provide better predictions. A final shortcoming of this thesis may be the final selection of models. In this thesis, five models were selected based on related work, however not all models were tested, due to time constraints and intended size of the thesis. A future study may be able to incorporate a different selection of machine learning algorithms or incorporate a newly designed algorithm with other properties that could provide even better predictions.

7 Conclusion

In this thesis, several machine learning algorithms were examined to predict the amount of time between the creation of a shift and its fulfilment. Among the machine learning algorithms investigated, RF based on RMSE and R^2 performed best. Moreover, important features were found that had a considerable impact on the amount of time needed to completely fill a shift. The RF could predict a large part of the variance based on the selected features even with large variability in the data distribution of the target variable. The predictions of the machine learning algorithms could reduce uncertainty for customers placing shifts in the future, which can lead to better shift placements and fulfilments. This thesis contributes to the scarce scientific work on time prediction in combination with an online job platforms. For future works in this research field, it is recommended to incorporate more data and explore new features to get better predictions.

8 Acknowledgements

I would like to thank Jesse Kirschner and David Telenga for providing the dataset and helping with any questions regarding the dataset and problem definition. I would also like to thank Boris Čule for his guidance during the thesis process.

References

Abraham, K. G., Haltiwanger, J. C., Sandusky, K., & Spletzer, J. R. (2018). Measuring the gig economy: current knowledge and open issues. *Working Paper Series*, 24950.

Aldrich, C., & Auret, L. (2010). Fault detection and diagnosis with random forest feature extraction and variable importance methods. *IFAC Proceedings Volumes*, *43*(9), 79–86. https://doi.org/10.3182/20100802-3-za-2014.00020.

Bellesia, F., Mattarelli, E., Bertolotti, F., & Sobrero, M. (2019). Platforms as entrepreneurial incubators? How online labor markets shape work identity. *Journal of Managerial Psychology*, *34*(4), 246–268. https://doi.org/10.1108/jmp-06-2018-0269.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/a:1010933404324

Brownlee, J. (2019, August 12). *Overfitting and Underfitting With Machine Learning Algorithms*. Machine Learning Mastery. https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/.

Brownlee, J. (2020, August 20). *How to Calculate Feature Importance With Python*. Machine Learning Mastery. https://machinelearningmastery.com/calculate-feature-importance-with-python/.

Codagnone, C., Abadie F., Biagi, F. (2016). The Future of Work in the 'Sharing Economy'. Market Efficiency and Equitable Opportunities or Unfair Precarisation? Institute for Prospective Technological Studies, JRC Science for Policy Report EUR 27913 EN, doi:10.2791/431485.

Curtis, C., Liu, C., Bollerman, T. J., & Pianykh, O. S. (2018). Machine learning for predicting patient wait times and appointment delays. *Journal of the American College of Radiology*, 15(9), 1310–1316. https://doi.org/10.1016/j.jacr.2017.08.021.

Doshi, S. (2020, August 3). *Various Optimization Algorithms For Training Neural Network*. Medium. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6.

Ekwaru, J. P., & Veugelers, P. J. (2018). The Overlooked Importance of Constants Added in Log Transformation of Independent Variables with Zero Values: A Proposed Approach for Determining an Optimal Constant. *Statistics in Biopharmaceutical Research*, *10*(1), 26–29. https://doi.org/10.1080/19466315.2017.1369900

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, *26*(2), 105–109. https://doi.org/10.3969/j.issn.1002-0829.2014.02.009

Grover, P. (2020, May 20). *Getting Deeper into Categorical Encodings for Machine Learning*. Medium. https://towardsdatascience.com/getting-deeper-into-categorical-encodings-for-machine-learning-2312acd347c8.

Heaton, J. T. (2005). *Introduction to Neural Networks with Java*. Heaton Research, Inc.

Hearst, M., Dumais, S., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and Their Applications*, 13(4), 18–28. https://doi.org/10.1109/5254.708428

Hocking, R. R. (1983). Developments in Linear Regression Methodology: 1959-1982. *Technometrics, 25*(3), 219–230. https://doi.org/10.2307/1268603

Johnson, K. (2021). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press.

Kässi, O., & Lehdonvirta, V. (2018). Online labour index: measuring the online gig economy for policy and research. Technological Forecasting & Social Change, 137, 241–248. https://doi.org/10.1016/j.techfore.2018.07.056.

Kokkodis, M., Papadimitriou, P., & Ipeirotis, P. G. (2015). Proceedings of the eighth acm international conference on web search and data mining. In *Hiring behavior models for online labor markets* (pp. 223–232). essay. https://doi.org/10.1145/2684822.2685299.

Kuo, Y. H., Chan, N. B., Leung, J. M. Y., Meng, H., So, A. M., Tsoi, K. K. F., & Graham, C. A. (2020). An integrated approach of machine learning and systems thinking for waiting time prediction in an emergency department. *International Journal of Medical Informatics*, 139, 104143–104143. https://doi.org/10.1016/j.ijmedinf.2020.104143.

Lewinson, E. (2019, August 26). *Explaining Feature Importance by example of a Random Forest*. Medium. https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e

Lin, W. C., Goldstein, I. H., Hribar, M. R., Sanders, D. S., & Chiang, M. F. (2020). Predicting Wait Times in Pediatric Ophthalmology Outpatient Clinic Using Machine Learning. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, *2019*, 1121–1128.

Martinez, O., Martinez, C., Parra, C. A., Rugeles, S., & Suarez, D. R. (2021). Machine learning for surgical time prediction. *Computer Methods and Programs in Biomedicine*, 208, 106220–106220. https://doi.org/10.1016/j.cmpb.2021.106220.

O'Farrell, R., & Montagnier, P. (2020). Measuring digital platform-mediated workers. *New Technology, Work and Employment*, 35(1), 130–144. https://doi.org/10.1111/ntwe.12155

Oshiro, Thais & Perez, Pedro & Baranauskas, José. (2012). How Many Trees in a Random Forest? *Lecture notes in computer science*. 7376. 10.1007/978-3-642-31537-4_13.

Płoński, P. (2020, June 29). *Random Forest Feature Importance Computed in 3 Ways with Python*. MLJAR. https://mljar.com/blog/feature-importance-in-random-forest/

Pongratz, H. J. (2018). Of crowds and talents: discursive constructions of global online labour. *New Technology, Work and Employment*, 33(1), 58–73. https://doi.org/10.1111/ntwe.12104.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In *Advances in neural information processing systems* (pp. 6638-6648).

Reed, R. D. (1999). *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press (MA).

Ronaghan, S. (2019, November 1). *The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark*. Medium. https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

Samba, A., Busnel, Y., Blanc, A., Dooze, P., & Simon, G. (2018). Predicting file downloading time in cellular network: large-scale analysis of machine learning approaches. *Computer Networks*, 145, 243–254. https://doi.org/10.1016/j.comnet.2018.09.002

Tharsanee, R.M., Soundariya, R.S., Kumar, A., Karthiga, M., Sountharrajan, S. (2021). 7 - Deep convolutional neural network–based image classification for COVID-19 diagnosis, Editor(s): Utku Kose, Deepak Gupta, Victor Hugo C. de Albuquerque, Ashish Khanna, *Data Science for COVID-19* (pp. 117-145). Academic Press, ISBN 9780128245361, https://doi.org/10.1016/B978-0-12-824536-1.00012-5

West, R. M. (2021). Best practice in statistics: The use of log transformation. *Annals of Clinical Biochemistry: International Journal of Laboratory Medicine*, 000456322110505. https://doi.org/10.1177/00045632211050531

Wiel, B., van der Werf, S. and van Kesteren, J. (2020, October 20). *Meting kluseconomie*. SEO Economisch Onderzoek. https://www.seo.nl/publicaties/meting-kluseconomie/

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology), 67*(2), 301–320. http://www.jstor.org/stable/3647580

Documentation and packages

Dplyr package - RDocumentation. (2022). Rdocumentation. https://www.rdocumentation.org/packages/dplyr/versions/0.7.8

Feature importances with a forest of trees. (2021). Scikit-Learn. https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

Grolemund, G., & Wickham, H. (2011). Dates and Times Made Easy withlubridate. *Journal of Statistical Software, 40(3)*. https://doi.org/10.18637/jss.v040.i03.

Permutation feature importance. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/permutation_importance.html

Sklearn.ensemble.RandomForestRegressor. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

Sklearn.linear_model.ElasticNet. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html.

Sklearn.linear_model.LinearRegression. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html.

Sklearn.model_selection.GridSearchCV. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Sklearn.model_selection.train_test_split. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Sklearn.svm.SVR. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR

Team, K. (2021). Keras documentation: The Sequential model. Keras. https://keras.io/guides/sequential_model/

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. doi: 10.21105/joss.01686. https://joss.theoj.org/papers/10.21105/joss.01686.

Appendix

Table 1

Description of the selected features

Feature name	Description
Shift_createdAt	The date and time the shift was created.
Shift_start	The starting time of the shift.
Shift_end	The end time of the shift.
Max_claims	The maximum number of flex workers a particular shift can have.
Hourly_wage	Pay per hour.
Fine_noShowJob	Penalty charge for a particular shift if you do not show up.
Job_function	Job title / function description.
Job_location	Workplace location.
Diff_time	The time difference between shift fulfilment and the creation of the shift.

Note. Diff_time is the target variable in this thesis.